

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Douglas L. Rollins § Group Art Unit: 2122
Serial No.: 09/272,845 §
Filed: March 19, 1999 § Examiner: Chuck O. Kendall
For: Software Module Update § Atty. Dkt. No.: MCT.0088US
(MUEI-0471.00/US)

Board of Patent Appeals & Interferences
Commissioner for Patents
Washington, D.C. 20231

#10
p. Cutrona
9-3-02
RECEIVED

AUG 15 2002

Technology Center 2100

APPEAL BRIEF

Dear Sir:

Applicant hereby appeals from the Final Rejection dated February 6, 2002, finally rejecting claims 1-44.

I. REAL PARTY IN INTEREST

The real party in interest is Micron Technology, Inc.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals of interferences.

III. STATUS OF THE CLAIMS

The application was originally filed with claims 1-40, and claims 41-44 were added by amendment. In an after final Reply, a request was made to cancel claims 33-40. The Advisory

08/19/2002 SCOTTON 00000005 09272845
01 FC:120 320.00 OP

Date of Deposit: August 6, 2002
I hereby certify under 37 CFR 1.8(a) that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated above and is addressed to the Board of Patent Appeals & Interferences, Commissioner for Patents, Washington, DC 20231.
Debra Cutrona
Debra Cutrona

Action did not indicate whether claims 33-40 have been canceled. However, because the cancellation of claims 33-40 narrows down the issues on appeal, it is assumed in this Appeal Brief that the amendment will be entered. Claims 1-32 and 41-44 have been finally rejected and are the subject of this appeal.

IV. STATUS OF AMENDMENTS

An amendment canceling claims 33-40 has not yet been entered (or at least an indication of the entering of the amendment has not been received by Applicant). There are no other unentered amendments.

V. SUMMARY OF THE INVENTION

Techniques (including methods and devices) to enable software module updates are described. The following embodiments, described in terms of techniques to update program applications and software driver modules, are illustrative only and are not to be considered limiting in any respect. Further, in the interest of clarity not all features of an actual implementation are described herein. It will be appreciated that the development of any actual implementation requires numerous programming decisions to achieve the developer's specific goals such as compliance with system-related and business-related constraints. Moreover, these decisions may be complex and time-consuming but would nevertheless be a routine undertaking for those of ordinary skill having the benefit of this disclosure. Specification, p. 2.

Referring to FIG. 1, a system that provides software module update capability is shown in accordance with one embodiment of the invention. Computer system 100 may be coupled to

update information 102 and update source 104 via communication link 106. Illustrative computer systems include general purpose (e.g., PCs) and special purpose (e.g., graphics workstations) computer systems. Update information 102 may be a database or other information storage facility that includes versioning information for various software modules. Update source 104 represents that location or locations at which software module updates are stored. Communication link 106 may be a modem connection, or a direct connection to a local area or wide area network. The form of communication link 106 is irrelevant to the invention and may, for example, employ copper wire, radio frequency or optical technologies. It will be understood that update information 102 and update source 104 may be co-located. It will be further understood that update source 104 may include multiple sites (e.g., various world wide web sites). Specification, pp. 2-3.

Update information 102 may include a variety of data for each software module that may be updated in accordance with the invention. For example, update information 102 may be organized as a database with one record for each available software module. (Software modules may include user programs such as word processing and graphics applications, or drivers to control either a hardware device (device driver) or another software subsystem.) Each record may include information representing one or more of the following: software module name; manufacturer's identification number, device identification number; device version number; BIOS version number; driver version number; date; an indication of what versions the current version is an update for; and a location of the identified module. Specification, p. 3.

Referring again to FIG. 1, computer system 100 may include processor 108, memory 110, and possibly add-in cards such as video and modem cards 112 and 114 respectively. Illustrative processors include the PENTIUM[®] family of processors and the 80X86 families of processors from Intel Corporation. Memory 110 may include volatile (e.g., dynamic random access memory) and/or non-volatile memory (e.g., non-volatile random access memory, magnetic or optical disk units). Add-in cards represent physical devices that augment the processing features of processor 108. Typically, an add-in card includes a programmable control device (e.g., a microprocessor, microcontroller, or a specially designed programmable state machine) and associated control memory (often referred to as BIOS). Specification, pp. 3-4.

As shown, memory 110 may include update routine 116 and one or more software programs 118. Update routine 116 represents one or more software program modules that works in conjunction with update information 102 and update source 104 to provide the software module update capability of the present invention. Programs 118 represents one or more user or operating system applications which may be updated by update routine 116. Specification, p. 4.

Referring to FIG. 2, a method to provide software module update capability is shown in accordance with one embodiment of the invention. Initially a user indicates they want to load a new program module or update an existing module (block 200). For example, a user may have purchased a new video controller add-in card and now wants to install the appropriate device drivers. Update routine 116 may receive this notification in any convenient manner. Specification, p. 4.

Following notification, update routine 116 obtains version information for the module being loaded/updated (block 202). If the module being updated is associated with a software application, version information may be obtained through standard queries to, or inspection of, one or more of computer system 100's system files. If the module being updated is associated with a hardware device (e.g., an add-in card), update routine 116 may interrogate the card directly to obtain one or more of the following: a device identifier value; a subsystem identifier value; a device version identifier value; the device's BIOS version identifier value; and the version number of any currently loaded device drives associated with the device. Specification, p. 4.

Identification of version information for a software application or a previously loaded device driver may be obtained through standard queries to, or inspection of, one or more of computer system 100's system files. By way of example, in a Microsoft Windows® operating system information about those driver routines and program applications that are loaded may be obtained from the Registry file. Direct interrogation of a physical device (e.g., video controller or network interface cards) to determine the device's version information is preferred over an inspection of system files such as the Windows® Registry file. Specification, pp. 4-5.

Following the acts of block 202, update routine 116 may communicate with update information 102 via communication link 106 to determine what program modules are the most current for the identified device or program (block 204). Next, update routine 116 indicates the most recent module (e.g., device driver or program) to the user (block 206) which may then be loaded from update source 104 in accordance with current techniques (block 208). The acts of

FIG. 2 may be initiated when a user begins installation of a new device or program or at any subsequent time specified by the user. Specification, p. 5.

In one embodiment, update routine 116 shows only the most current modules for loading to the user by comparing the version obtained via interrogating system 100 (the acts of block 202) and the information obtained from update information 102 (the acts of block 204). It will be recognized that the update module (i.e., that module which will update the module identified during the acts of blocks 100 and 102) may be obtained from any location (update source 104) in communication with computer system 100. For example, a video controller add-in card's device driver may be located at the video card's manufacturer's website or at the website of an original equipment manufacturer who assembled and sold the video card with computer system 100. The update module may also be local to computer system 100. Specification, p. 5.

One benefit of the invention is that it allows a user to maintain their system in an up to date state without requiring them to have or obtain detailed technical information about system components. Another benefit of the invention is that update routine 116 may also prevent use of inappropriate updates. For example, if a user's network controller card is not capable of using the chronologically most recent update, but instead may only use those driver versions before a specified date, update routine 116 may obtain this information from update information 102 and display only the most "relevant" current version to the user for loading during the acts of block 206 and 208. Yet another benefit of the invention is that a user does not have to know where a needed update module is located. In one embodiment the user only needs to know where or how to establish communication with update information 102. In another embodiment, the location

of update information 102 may be predetermined by the computer system vendor. Still another benefit of the invention is the automatic nature of update routine 116. That is, update routine 116 determines not only what driver updates are appropriate but may also (automatically or following user authorization) retrieve the update modules/software and install them on a user's machine. Thus, in contrast to prior art update techniques the user does not have to know: the technical details of their system's current software load (e.g., drivers and application programs); explicit location(s) where various update modules may be located; or how to affirmatively download the various update routines. Specification, pp. 5-6.

Various changes in the materials, components, circuit elements, as well as in the details of the illustrated operational method are possible without departing from the scope of the claims. For instance, acts in accordance with FIG. 2 may be performed by a programmable control device executing instructions organized into a program module (e.g., update routine 116). A programmable control device may be a single computer processor (e.g., processor 108), a plurality of computer processors coupled by a communications link, or a custom designed state machine. Custom designed state machines may be embodied in a hardware device such as a printed circuit board comprising discrete logic, integrated circuits, specially designed application specific integrated circuits (ASICs), or field programmable gate array devices. Storage devices suitable for tangibly embodying program instructions include all forms of non-volatile memory including, but not limited to: semiconductor memory devices such as EPROM, EEPROM, and flash devices; magnetic disks (fixed, floppy, and removable); other magnetic media such as tape; and optical media such as CD-ROM disks. Specification, p. 6.

VI. ISSUES

- A. Can claims 1-16, 41 and 42 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness?
- B. Can claims 17-32, 43 and 44 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness?

VII. GROUPING OF THE CLAIMS

Claims 1-16, 41 and 42 can be grouped together; and claims 17-32, 43 and 44 can be grouped together.

VIII. ARGUMENT

All claims are allowable for the reasons set forth below.

- A. Can claims 1-16, 41 and 42 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness?

The method of independent claim 1 includes identifying a first version of a software module. The software module is installed on a computer system and is associated with circuitry of the computer system. The method includes identifying a second version of the software module and automatically determining whether the second version is more current than the first version and whether the second version is compatible with the circuitry. The method further includes indicating the result of the determination.

The Examiner rejects independent claim 1 under 35 U.S.C. § 103(a) in view of two references: U.S. Patent No. 5,974,494 (herein referred to as "Furner") and U.S. Patent No. 5,974,454 (herein referred to as "Apfel"). Furner is generally directed to a system that automatically detects the installation of a hardware device, and when the hardware device is

installed, the system selects a driver that can support the hardware device. For example, *see* Furner, 3:28-31. Furner also describes a weighted value system for selecting the particular driver based on the driver's ability to support the hardware. Furner, 25:26-59. However, Furner neither teaches nor suggests determining whether a second version of a driver is more current than a first version of a driver. Thus, Furner teaches determining the compatibility of a newly installed piece of hardware to drivers in response to the detection of the new hardware but does not disclose determining whether a second version of a software module is more current than a first version of the software module.

Apfel teaches a method and system for installing and updating program module components. In particular, Apfel teaches determining whether a new version of a particular software package is available and if so, Apfel discloses updating the software on the computer. Apfel, however, does not disclose determining whether a second version of a software module is compatible with circuitry that is associated with the software module.

To establish a *prima facie* case of obviousness, there must be a suggestion or motivation either in the references themselves or in the general level of skill in the art to support the combination or modification. M.P.E.P. § 2143. The Examiner does not allege that that a suggestion or motivation exists in the general level in skill in the art for the combination of Apfel and Furner. However, the Examiner reaches the unsupported conclusion that such a suggestion or motivation exists without specifically pointing out where such a suggestion or motivation is present in the cited references.

In this manner, "obviousness cannot be predicated on what is unknown." *In re Spormann*, 363 F.2d 444, 448, 150 USPQ 449, 452 (CCPA 1966). Without such a specific cite, a *prima facie* case of obviousness has not been established. *Ex parte Gambogi*, 62 USPQ2d 1209, 1212 (Bd. Pat. App. & Int. 2001); *In re Rijckaert*, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993). Without a specific cite to the language that supplies the alleged suggestion or motivation, Applicant has no opportunity to review the language and possibly present arguments why the language does not supply the suggestion or motivation. Thus, because Applicant has not had the opportunity to contest the alleged suggestion or motivation, "the burden to rebut a rejection of obviousness does not arise until a *prima facie* case of obviousness has been established." *In re Rijckaert*, 28 USPQ2d at 1957.

Not only does Apfel fail to supply the suggestion or motivation for the combination, Apfel teaches away from the § 103 combination. In this manner, Apfel is not concerned whether the most recent software update is compatible with the associated circuitry on Apfel's computer, as the update occurs regardless of whether a determination of this compatibility is made. Thus, Apfel teaches away from its combination with Furner. References cannot be combined where a reference teaches away from their combination. *In re Grasselli*, 713 F.2d, 731, 743, 218 USPQ 769, 779 (Fed. Cir. 1983); M.P.E.P. § 2145X(D)(2).

Additionally, Furner fails to provide a suggestion or motivation for the § 103 combination. In this manner, Furner teaches determining compatibility to a particular program in response to the detection of new hardware, not in response to updating a particular software

driver, for example. Thus, one skilled in the art would not have been motivated to apply Furner's teaching to Apfel.

Therefore, neither Apfel, Furner, nor any other reference that was cited by the Examiner provides a suggestion or motivation to combine Apfel and Furner.

Thus, the rejections of claims 1-16, 41 and 42 are improper and should be reversed.

B. Can claims 17-32, 43 and 44 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness?

The program storage device of independent claim 17 is readable by a programmable control device and includes instructions that are stored on the program storage device for causing the programmable control device to identify a first version of a software module. The software module is installed on the computer system and is associated with the circuitry of the computer system. The instructions cause the programmable control device to identify a second version of the software module, automatically determine whether the second version is more current than the first version and whether the second version is compatible with the circuitry and indicate the result of this determination.

The Examiner applies Apfel and Furner in rejecting claims 17-32, 43 and 44 under 35 U.S.C. § 103(a). However, the Examiner fails to establish a *prima facie* case of obviousness, as the Examiner does not specifically point out where such a suggestion or motivation exists to combine Apfel and Furner. Alone, neither one of these references teaches all of the limitations of claim 26. Furthermore, as set forth above in Issue A, neither Apfel, Furner, nor any other reference cited by the Examiner provides a suggestion or motivation for the combination.

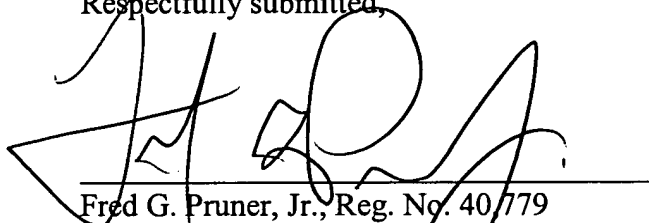
Because "obviousness cannot be predicated on what is unknown," a *prima facie* case of obviousness has not been established. *In re Spormann*, 150 USPQ at 452.

Thus, the rejections of claims 17-32, 43 and 44 are improper and should be reversed.

IX. CONCLUSION

The Assignee requests that each of the final rejections be reversed and that the claims subject to this appeal be allowed to issue.

Respectfully submitted,



Fred G. Pruner, Jr., Reg. No. 40,779
TROP, PRUNER & HU, P.C.
8554 Katy Freeway, Suite 100
Houston, TX 77024-1805
713/468-8880 [Phone]
713/468-8883 [Facsimile]

Date: August 6, 2002



21906

PATENT TRADEMARK OFFICE

APPENDIX OF CLAIMS

The claims on appeal are:

1. A method comprising:

identifying a first version of a software module, the software module being installed on a computer system and being associated with circuitry of the computer system;

identifying a second version of the software module;

automatically determining whether the second version is more current than the first version and whether the second version is compatible with the circuitry; and

indicating the result of the determination.
2. The method of claim 1, further comprising obtaining that version of the software module determined to be the most current version.
3. The method of claim 2, further comprising loading the obtained version of the software module.
4. The method of claim 1, wherein the act of identifying a first version of the software module comprises communicating with a physical device associated with the software module.
5. The method of claim 4, wherein the act of communicating comprises determining an identifier value of the physical device.

6. The method of claim 5, further comprising determining a subsystem identifier value of the physical device.

7. The method of claim 4, wherein the act of communicating comprises determining a basic input-output system version identifier value of the physical device.

8. The method of claim 1, wherein the act of identifying a second version of the software module comprises communicating with an update information source.

9. The method of claim 8, wherein the act of communicating further comprises:
identifying the software module to the update information source;
receiving, from the update information source, an indication of the second version of the software module.

10. The method of claim 8, wherein the act of communicating comprises communicating by a modem.

11. The method of claim 8, wherein the act of communicating comprises communicating by a computer network.

12. The method of claim 8, wherein the act of communicating comprises communicating with a database server device.

13. The method of claim 1, wherein the act of determining comprises comparing at least one characteristic of the first identified first version of the software module with the same characteristic of the second identified first version of the software module.

14. The method of claim 1, wherein the act of indicating comprises visually displaying an indication of the software module determined to be the most current version to a user.

15. The method of claim 2, wherein the act of obtaining comprises retrieving that version of the software module determined to be the most current version from an update source.

16. The method of claim 15, wherein the act of retrieving comprises retrieving from an update source that is physically distinct from the location of the first identified version of the software module.

17. A program storage device, readable by a programmable control device, comprising:

instructions stored on the program storage device for causing the programmable control device to

identify a first version of a software module, the software module being installed on a computer system and being associated with the circuitry of the computer system;

identify a second version of the software module;

automatically determine whether the second version is more current than the first version and whether the second version is compatible with the circuitry; and
indicate the result of the determination.

18. The program storage device of claim 17, further comprising instructions to obtain that version of the software module determined to be most current version.

19. The program storage device of claim 18, further comprising instructions to load the obtained version of the software module.

20. The program storage device of claim 17, wherein the instructions to identify a first version of the software module comprise instructions to communicate with a physical device associated with the software module.

21. The program storage device of claim 20, wherein the instructions to communicate comprise instructions to determine an identifier value of the physical device.

22. The program storage device of claim 21, further comprising instructions to determine a subsystem identifier value of the physical device.

23. The program storage device of claim 20, wherein the instructions to communicate comprise instructions to determine a basic input-output system version identifier value of the physical device.

24. The program storage device of claim 17, wherein the instructions to identify a second version of the software module comprise instructions to communicate with an update information source.

25. The program storage device of claim 24, wherein the instructions to communicate further comprise instructions to:

identify the software module to the update information source;

receive, from the update information source, an indication of the second version of the software module.

26. The program storage device of claim 24, wherein the instructions to communicate comprise instructions to communicate by a modem.

27. The program storage device of claim 24, wherein the instructions to communicate comprise instructions to communicate by a computer network.

28. The program storage device of claim 24, wherein the instructions to communicate comprise instructions to communicate with a database server device.

29. The program storage device of claim 17, wherein the instructions to determine comprise instructions to compare at least one characteristic of the first identified first version of the software module with the same characteristic of the second identified first version of the software module.

30. The program storage device of claim 17, wherein the instructions to indicate comprise instructions to visually display an indication of the software module determined to be the most current version to a user.

31. The program storage device of claim 18, wherein the instructions to obtain comprise instructions to retrieve that version of the software module determined to be the most current version from an update source.

32. The program storage device of claim 31, wherein the instructions to retrieve comprise instructions to retrieve from an update source that is physically distinct from the location of the first identified version of the software module.

41. The method of claim 1, wherein the determining comprises:
determining a date associated with the first version, the date establishing compatibility between the second and first versions.

42. The method of claim 1, wherein the circuitry comprises an add-in card.

43. The program storage device of claim 17, further comprising instructions to cause the programmable control device to determine a date associated with the first version, the date establishing compatibility between the first and second versions.

44. The programmable storage device of claim 17, wherein the circuitry comprises an add-in card.

TRANSMITTAL OF APPEAL BRIEF (Large Entity)

Docket No.
MCT.0088US

In Re Application Of: Douglas L. Rollins

Serial No.
09/272,845Filing Date
March 19, 1999Examiner
Chuck O. KendallGroup Art Unit
2122

Invention: Software Module Update

TO THE ASSISTANT COMMISSIONER FOR PATENTS:

Transmitted herewith in triplicate is the Appeal Brief in this application, with respect to the Notice of Appeal filed on June 6, 2002

The fee for filing this Appeal Brief is: \$320.00

- ☒ A check in the amount of the fee is enclosed.
- ☐ The Commissioner has already been authorized to charge fees in this application to a Deposit Account. A duplicate copy of this sheet is enclosed.
- ☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 20-1504
A duplicate copy of this sheet is enclosed.

RECEIVED
AUG 13 PM 2:52
BOARD OF PATENT APPEALS
AND INTERFERENCES

Dated: August 6, 2002

Signature

Fred G. Pruner, Jr., Reg. No. 40,779
Trop, Pruner & Hu, P.C.
8554 Katy Freeway, Suite 100
Houston, Texas 77024
(713) 468-8880 [Phone]
(713) 468-8883 [Fax]

RECEIVED

AUG 15 2002

Technology Center 2100



21906

PATENT TRADEMARK OFFICE

I certify that this document and fee is being deposited on 08/06/02 with the U.S. Postal Service as first class mail under 37 C.F.R. 1.8 and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Signature of Person Mailing Correspondence

Debra Cutrona

Typed or Printed Name of Person Mailing Correspondence

CC: